

```
entry:
  call void @llvm.dbg.value(metadata ptr %a, metadata !21, metadata
  ... !DIExpression()), !dbg !27
  call void @llvm.dbg.value(metadata ptr %b, metadata !22, metadata
  ... !DIExpression()), !dbg !27
  call void @llvm.dbg.value(metadata ptr %c, metadata !23, metadata
  ... !DIExpression()), !dbg !27
  call void @llvm.dbg.value(metadata %n, metadata !24, metadata
  ... !DIExpression()), !dbg !27
  call void @llvm.dbg.value(metadata i32 0, metadata !25, metadata
  ... !DIExpression()), !dbg !28
  %cmp11 = icmp sgt i32 %n, 0, !dbg !29
  br i1 %cmp11, label %for.body.preheader, label %for.cond.cleanup, !dbg !31

T                                         F
```

```
for.body.preheader:
  %wide.trip.count = zext i32 %n to i64, !dbg !29
  br label %for.body.init.1, !dbg !31
```

```
for.body.init.1:
  call void @llvm.dbg.value(metadata i64 0, metadata !25, metadata
  ... !DIExpression()), !dbg !28
  %rem15.init.1 = and i64 0, 1, !dbg !32
  %cmp1.not.init.1 = icmp eq i64 %rem15.init.1, 0, !dbg !32
  br label %for.inc.init.1
```

```
for.inc.init.1:
  %indvars.iv.next.init.1 = add nuw nsw i64 0, 1, !dbg !35
  call void @llvm.dbg.value(metadata i64 %indvars.iv.next.init.1, metadata
  ... !25, metadata !DIExpression()), !dbg !28
  br label %for.body.headerCopy.1.init.1
```

```
for.body.headerCopy.1.init.1:
  call void @llvm.dbg.value(metadata i64 %indvars.iv.next.init.1, metadata
  ... !25, metadata !DIExpression()), !dbg !28
  %rem15.headerCopy.1.init.1 = and i64 %indvars.iv.next.init.1, 1, !dbg !32
  %cmp1.not.headerCopy.1.init.1 = icmp eq i64 %rem15.headerCopy.1.init.1, 0,
  ... !dbg !32
  br label %for.inc.latchCopy.1.init.1
```

```
for.inc.latchCopy.1.init.1:
  %indvars.iv.next.latchCopy.1.init.1 = add nuw nsw i64
  ... %indvars.iv.next.init.1, 1, !dbg !35
  call void @llvm.dbg.value(metadata i64 %indvars.iv.next.latchCopy.1.init.1,
  ... metadata !25, metadata !DIExpression()), !dbg !28
  br label %for.body.headerCopy.1.2.init.1
```

```
for.body.headerCopy.1.2.init.1:
  call void @llvm.dbg.value(metadata i64 %indvars.iv.next.latchCopy.1.init.1,
  ... metadata !25, metadata !DIExpression()), !dbg !28
  %rem15.headerCopy.1.2.init.1 = and i64 %indvars.iv.next.latchCopy.1.init.1,
  ... 1, !dbg !32
  %cmp1.not.headerCopy.1.2.init.1 = icmp eq i64 %rem15.headerCopy.1.2.init.1,
  ... 0, !dbg !32
  br label %for.inc.latchCopy.1.2.init.1
```

```
for.inc.latchCopy.1.2.init.1:
  %indvars.iv.next.latchCopy.1.2.init.1 = add nuw nsw i64
  ... %indvars.iv.next.latchCopy.1.2.init.1, 1, !dbg !35
  %exitcond.not.latchCopy.1.2.3.init.1 = icmp eq i64
  ... %indvars.iv.next.latchCopy.1.2.init.1, metadata !25, metadata
  ... !DIExpression()), !dbg !28
  br label %for.body.headerCopy.1.2.3.init.1
```

```
for.body.headerCopy.1.2.3.init.1:
  call void @llvm.dbg.value(metadata i64
  ... %indvars.iv.next.latchCopy.1.2.init.1, metadata !25, metadata
  ... !DIExpression()), !dbg !28
  %rem15.headerCopy.1.2.3.init.1 = and i64
  ... %indvars.iv.next.latchCopy.1.2.init.1, 1, !dbg !32
  %cmp1.not.headerCopy.1.2.3.init.1 = icmp eq i64
  ... %rem15.headerCopy.1.2.3.init.1, 0, !dbg !32
  br label %for.inc.latchCopy.1.2.3.init.1
```

```
for.inc.latchCopy.1.2.3.init.1:
  %indvars.iv.next.latchCopy.1.2.3.init.1 = add nuw nsw i64
  ... %indvars.iv.next.latchCopy.1.2.init.1, 1, !dbg !35
  call void @llvm.dbg.value(metadata i64
  ... %indvars.iv.next.latchCopy.1.2.3.init.1, metadata !25, metadata
  ... !DIExpression()), !dbg !28
  %0 = sub i64 %wide.trip.count, 3, !dbg !29
  %exitcond.not.latchCopy.1.2.3.init.1 = icmp eq i64
  ... %indvars.iv.next.latchCopy.1.2.3.init.1, %0, !dbg !29
  br label %for.body.init.2
```

```
for.body.init.2:
  call void @llvm.dbg.value(metadata i64 0, metadata !25, metadata
  ... !DIExpression()), !dbg !28
  %rem15.init.2 = and i64 %indvars.iv.next.latchCopy.1.2.3.init.1, 1, !dbg !32
  %cmp1.not.init.2 = icmp eq i64 %rem15.init.2, 0, !dbg !32
  br label %for.inc.init.2
```

```
for.inc.init.2:
  %indvars.iv.next.init.2 = add nuw nsw i64
  ... %indvars.iv.next.latchCopy.1.2.3.init.1, 1, !dbg !35
  call void @llvm.dbg.value(metadata i64 %indvars.iv.next.init.2, metadata
  ... !25, metadata !DIExpression()), !dbg !28
  br label %for.body.headerCopy.1.init.2
```

```
for.body.headerCopy.1.init.2:
  call void @llvm.dbg.value(metadata i64 %indvars.iv.next.init.2, metadata
  ... !25, metadata !DIExpression()), !dbg !28
  %rem15.headerCopy.1.init.2 = and i64 %indvars.iv.next.init.2, 1, !dbg !32
  %cmp1.not.headerCopy.1.init.2 = icmp eq i64 %rem15.headerCopy.1.init.2, 0,
  ... !dbg !32
  br label %for.inc.latchCopy.1.init.2
```

```
for.inc.latchCopy.1.init.2:
  %indvars.iv.next.latchCopy.1.init.2 = add nuw nsw i64
  ... %indvars.iv.next.init.2, 1, !dbg !35
  call void @llvm.dbg.value(metadata i64 %indvars.iv.next.latchCopy.1.init.2,
  ... metadata !25, metadata !DIExpression()), !dbg !28
  br label %for.body.headerCopy.1.2.init.2
```

```
for.body.headerCopy.1.2.init.2:
  call void @llvm.dbg.value(metadata i64 %indvars.iv.next.latchCopy.1.init.2,
  ... metadata !25, metadata !DIExpression()), !dbg !28
  %rem15.headerCopy.1.2.init.2 = and i64 %indvars.iv.next.latchCopy.1.init.2,
  ... 1, !dbg !32
  %cmp1.not.headerCopy.1.2.init.2 = icmp eq i64 %rem15.headerCopy.1.2.init.2,
  ... 0, !dbg !32
  br label %for.inc.latchCopy.1.2.init.2
```

```
for.inc.latchCopy.1.2.init.2:
  %indvars.iv.next.latchCopy.1.2.init.2 = add nuw nsw i64
  ... %indvars.iv.next.latchCopy.1.2.init.2, 1, !dbg !35
  call void @llvm.dbg.value(metadata i64
  ... %indvars.iv.next.latchCopy.1.2.init.2, metadata !25, metadata
  ... !DIExpression()), !dbg !28
  br label %for.body.headerCopy.1.2.3.init.2
```

```
for.body.headerCopy.1.2.3.init.2:
  call void @llvm.dbg.value(metadata i64
  ... %indvars.iv.next.latchCopy.1.2.init.2, metadata !25, metadata
  ... !DIExpression()), !dbg !28
  %rem15.headerCopy.1.2.3.init.2 = and i64
  ... %indvars.iv.next.latchCopy.1.2.init.2, 1, !dbg !32
  %cmp1.not.headerCopy.1.2.3.init.2 = icmp eq i64
  ... %rem15.headerCopy.1.2.3.init.2, 0, !dbg !32
  br label %for.inc.latchCopy.1.2.3.init.2
```

```
for.inc.latchCopy.1.2.3.init.2:
  %indvars.iv.next.latchCopy.1.2.3.init.2 = add nuw nsw i64
  ... %indvars.iv.next.latchCopy.1.2.init.2, 1, !dbg !35
  call void @llvm.dbg.value(metadata i64
  ... %indvars.iv.next.latchCopy.1.2.3.init.2, metadata !25, metadata
  ... !DIExpression()), !dbg !28
  %1 = sub i64 %wide.trip.count, 3, !dbg !29
  %exitcond.not.latchCopy.1.2.3.init.2 = icmp eq i64
  ... %indvars.iv.next.latchCopy.1.2.3.init.2, %1, !dbg !29
  %2 = insertelement <vscale x 4 x i1> undef, i1 %cmp1.not.init.1, i64 0, !dbg
  ... !31
  %3 = insertelement <vscale x 4 x i1> %2, i1 %cmp1.not.headerCopy.1.init.1,
  ... i64 1, !dbg !31
  %4 = insertelement <vscale x 4 x i1> %3, i1 %cmp1.not.headerCopy.1.2.init.1,
  ... i64 2, !dbg !31
  %5 = insertelement <vscale x 4 x i1> %4, i1
  ... %cmp1.not.headerCopy.1.2.3.init.1, i64 3, !dbg !31
  %6 = insertelement <vscale x 4 x i1> undef, i1 %cmp1.not.init.2, i64 0, !dbg
  ... !31
  %7 = insertelement <vscale x 4 x i1> %6, i1 %cmp1.not.headerCopy.1.init.2,
  ... i64 1, !dbg !31
  %8 = insertelement <vscale x 4 x i1> %7, i1 %cmp1.not.headerCopy.1.2.init.2,
  ... i64 2, !dbg !31
  %9 = insertelement <vscale x 4 x i1> %8, i1
  ... %cmp1.not.headerCopy.1.2.3.init.2, i64 3, !dbg !31
  %10 = call <vscale x 4 x i32> @llvm.aarch64.sve.index.nxv4i32(i32 0, i32 1),
  ... !dbg !31
  %11 = call <vscale x 4 x i32> @llvm.aarch64.sve.index.nxv4i32(i32 4, i32 1),
  ... !dbg !31
  br i1 %exitcond.not.latchCopy.1.2.3.init.2, label
  ... %for.cond.cleanup.loopexit, label %for.body, !dbg !31, !llvm.loop !36

T                                         F
```

```
for.body:
  %12 = phi i64 [ %indvars.iv.next.latchCopy.1.2.3, %new.latch ], [ 0,
  ... %for.inc.latchCopy.1.2.3.init.2 ]
  %13 = phi <vscale x 4 x i32> [ %72, %new.latch ], [ %10,
  ... %for.inc.latchCopy.1.2.3.init.2 ]
  %14 = phi <vscale x 4 x i1> [ %73, %new.latch ], [ %5,
  ... %for.inc.latchCopy.1.2.3.init.2 ]
  %15 = phi <vscale x 4 x i32> [ %74, %new.latch ], [ %11,
  ... %for.inc.latchCopy.1.2.3.init.2 ]
  %16 = phi <vscale x 4 x i1> [ %75, %new.latch ], [ %9,
  ... %for.inc.latchCopy.1.2.3.init.2 ]
  call void @llvm.dbg.value(metadata i64 0, metadata !25, metadata
  ... !DIExpression()), !dbg !28
  %rem15 = and i64 %12, 1, !dbg !32
  %cmp1.not = icmp eq i64 %rem15, 0, !dbg !32
  br label %for.inc
```

```
for.inc:
  %indvars.iv.next = add nuw nsw i64 %12, 1, !dbg !35
  call void @llvm.dbg.value(metadata i64 %indvars.iv.next, metadata !25,
  ... metadata !DIExpression()), !dbg !28
  br label %for.body.headerCopy.1
```

```
for.body.headerCopy.1:
  call void @llvm.dbg.value(metadata i64 %indvars.iv.next, metadata !25,
  ... metadata !DIExpression()), !dbg !28
  %rem15.headerCopy.1 = and i64 %indvars.iv.next, 1, !dbg !32
  %cmp1.not.headerCopy.1 = icmp eq i64 %rem15.headerCopy.1, 0, !dbg !32
  br label %for.inc.latchCopy.1
```

```
for.inc.latchCopy.1:
  %indvars.iv.next.latchCopy.1 = add nuw nsw i64 %indvars.iv.next, 1, !dbg !35
  call void @llvm.dbg.value(metadata i64 %indvars.iv.next.latchCopy.1,
  ... metadata !25, metadata !DIExpression()), !dbg !28
  br label %for.body.headerCopy.1.2
```

```
for.body.headerCopy.1.2:
  call void @llvm.dbg.value(metadata i64 %indvars.iv.next.latchCopy.1,
  ... metadata !25, metadata !DIExpression()), !dbg !28
  %rem15.headerCopy.1.2 = and i64 %indvars.iv.next.latchCopy.1, 1, !dbg !32
  %cmp1.not.headerCopy.1.2 = icmp eq i64 %rem15.headerCopy.1.2, 0, !dbg !32
  br label %for.inc.latchCopy.1.2
```

```
for.inc.latchCopy.1.2:
  %indvars.iv.next.latchCopy.1.2 = add nuw nsw i64
  ... %indvars.iv.next.latchCopy.1, 1, !dbg !35
  call void @llvm.dbg.value(metadata i64 %indvars.iv.next.latchCopy.1.2,
  ... metadata !25, metadata !DIExpression()), !dbg !28
  br label %for.body.headerCopy.1.2.3
```

```
for.body.headerCopy.1.2.3:
  call void @llvm.dbg.value(metadata i64 %indvars.iv.next.latchCopy.1.2,
  ... metadata !25, metadata !DIExpression()), !dbg !28
  %rem15.headerCopy.1.2.3 = and i64 %indvars.iv.next.latchCopy.1.2, 1, !dbg !32
  %cmp1.not.headerCopy.1.2.3 = icmp eq i64 %rem15.headerCopy.1.2.3, 0, !dbg !32
  br label %for.inc.latchCopy.1.2.3
```

```
for.inc.latchCopy.1.2.3:
  %indvars.iv.next.latchCopy.1.2.3 = add nuw nsw i64
  ... %indvars.iv.next.latchCopy.1.2.3, 1, !dbg !35
  call void @llvm.dbg.value(metadata i64 %indvars.iv.next.latchCopy.1.2.3,
  ... metadata !25, metadata !DIExpression()), !dbg !28
  %71 = sub i64 %wide.trip.count, 3, !dbg !29
  %exitcond.not.latchCopy.1.2.3 = icmp eq i64
  ... %indvars.iv.next.latchCopy.1.2.3, %71, !dbg !29
  br i1 %exitcond.not.latchCopy.1.2.3, label %for.cond.cleanup.loopexit, label
  ... %permute.decision, !dbg !31, !llvm.loop !36

T                                         F
```

```
for.cond.cleanup.loopexit:
  br label %for.cond.cleanup, !dbg !40
```

```
permute.decision:
  %17 = call <vscale x 4 x i1> @llvm.aarch64.sve.ptrne.nxv4i1(i32 4),
  ... %18 = call i64 @llvm.aarch64.sve.cntp.nxv4i1<vscale x 4 x i1> %17, <vscale
  ... x 4 x i1> %14
  %19 = call i64 @llvm.aarch64.sve.cntp.nxv4i1<vscale x 4 x i1> %17, <vscale
  ... x 4 x i1> %16
  %20 = add i64 %18, %19
  %21 = icmp uge i64 %20, 4
  br i1 %21, label %lane.gather, label %linearized

T                                         F
```

```
lane.gather:
  %22 = call <vscale x 4 x i1> @llvm.aarch64.sve.ptrne.nxv4i1(i32 4)
  %23 = call <vscale x 4 x i32> @llvm.aarch64.sve.compact.nxv4i32<vscale x 4
  ... x i1> %14, <vscale x 4 x i32> %13
  %24 = call <vscale x 4 x i32> @llvm.aarch64.sve.compact.nxv4i32<vscale x 4
  ... x i1> %16, <vscale x 4 x i32> %15
  %25 = xor <vscale x 4 x i1> %14, shufflevector (<vscale x 4 x i1>
  ... poison, <vscale x 4 x i32> zeroinitializer)
  %26 = xor <vscale x 4 x i1> %16, shufflevector (<vscale x 4 x i1>
  ... poison, <vscale x 4 x i32> zeroinitializer)
  %27 = call <vscale x 4 x i32> @llvm.aarch64.sve.compact.nxv4i32<vscale x 4
  ... x i1> %25, <vscale x 4 x i32> %13
  %28 = call <vscale x 4 x i32> @llvm.aarch64.sve.compact.nxv4i32<vscale x 4
  ... x i1> %26, <vscale x 4 x i32> %15
  %29 = call i64 @llvm.aarch64.sve.cntp.nxv4i1<vscale x 4 x i1> %22, <vscale
  ... x 4 x i1> %14
  %30 = call <vscale x 4 x i1> @llvm.aarch64.sve.whilelt.nxv4i1.i64(i64 0, i64
  ... %32)
  %31 = call <vscale x 4 x i32> @llvm.aarch64.sve.splice.nxv4i32<vscale x 4 x
  ... i1> %30, <vscale x 4 x i32> %23, <vscale x 4 x i32> %31,
  %32 = call i64 @llvm.aarch64.sve.cntp.nxv4i1<vscale x 4 x i1> %22, <vscale
  ... x 4 x i1> %16
  %33 = call <vscale x 4 x i1> @llvm.aarch64.sve.whilelt.nxv4i1.i64(i64 0, i64
  ... %32)
  %34 = call <vscale x 4 x i32> @llvm.aarch64.sve.splice.nxv4i32<vscale x 4 x
  ... i1> %33, <vscale x 4 x i32> %24, <vscale x 4 x i32> %28
  %35 = call i64 @llvm.aarch64.sve.cntp.nxv4i1<vscale x 4 x i1> %22, <vscale
  ... x 4 x i1> %16
  %36 = call <vscale x 4 x i1> @llvm.aarch64.sve.whilelt.nxv4i1.i64(i64 0, i64
  ... %35)
  %37 = call <vscale x 4 x i32> @llvm.aarch64.sve.whilelt.nxv4i32<vscale x 4 x
  ... i1> %36, <vscale x 4 x i32> %27, <vscale x 4 x i32> %31,
  %38 = xor <vscale x 4 x i1> %36, shufflevector (<vscale x 4 x i1>
  ... insertelement (<vscale x 4 x i1> poison, i1 true, i32 0), <vscale x 4 x i1>
  ... poison, <vscale x 4 x i32> zeroinitializer)
  %39 = call i64 @llvm.aarch64.sve.cntp.nxv4i1<vscale x 4 x i1> %22, <vscale
  ... x 4 x i1> %36
  %40 = sub i64 %32, %35
  %41 = add i64 %40, %39
  %42 = call <vscale x 4 x i1> @llvm.aarch64.sve.whilelt.nxv4i1.i64(i64 0, i64
  ... %41)
  %43 = xor <vscale x 4 x i1> %42, shufflevector (<vscale x 4 x i1>
  ... insertelement (<vscale x 4 x i1> poison, i1 true, i32 0), <vscale x 4 x i1>
  ... poison, <vscale x 4 x i32> zeroinitializer)
  %44 = and <vscale x 4 x i1> %38, %42
  %45 = and <vscale x 4 x i1> %38, %42
  %46 = and <vscale x 4 x i1> %45, %43
  %47 = or <vscale x 4 x i1> %44, %46
  br label %if.then
```

```
if.then:
  %arrayidx = getelementptr inbounds i32, ptr %a, i64 %12, !dbg !41
  %arrayidx3 = getelementptr inbounds i32, ptr %b, i64 %12, !dbg !47
  %arrayidx5 = getelementptr inbounds i32, ptr %c, i64 %12, !dbg !49
  %60 = call <vscale x 4 x i1> @llvm.aarch64.sve.ptrne.nxv4i1(i32 4), !dbg !51
  %61 = call i64 @llvm.aarch64.sve.cntp.nxv4i1(i32 4), !dbg !51
  ... @llvm.aarch64.sve.ldl.gather.sxtwi.index.nxv4i32<vscale x 4 x i1> %60, ptr
  ... %arrayidx, <vscale x 4 x i32> %31, !dbg !51
  %62 = call <vscale x 4 x i32>
  ... @llvm.aarch64.sve.ldl.gather.sxtwi.index.nxv4i32<vscale x 4 x i1> %60, ptr
  ... %arrayidx3, <vscale x 4 x i32> %31, !dbg !51
  %63 = call <vscale x 4 x i32> @llvm.aarch64.sve.mul.nxv4i32<vscale x 4 x
  ... i1> %60, <vscale x 4 x i32> %62, <vscale x 4 x i32> %61, !dbg !51
  call void @llvm.aarch64.sve.st.scatter.sxtwi.index.nxv4i32<vscale x 4 x i1> %63,
  ... <vscale x 4 x i1> %60, ptr %arrayidx5, <vscale x 4 x i32> %31, !dbg !51
  %64 = zext <vscale x 4 x i32> %31 to <vscale x 4 x i64>, !dbg !51
  call void @llvm.aarch64.sve.st.scatter.sxtwi.index.nxv4i32<vscale x 4 x i1> %64,
  ... <vscale x 4 x i32> %62, <vscale x 4 x i32> %61, !dbg !51
  %65 = insertelement <vscale x 4 x i1> undef, i1 %cmp1.not, i1 false, !dbg !51
  %66 = insertelement <vscale x 4 x i1> %65, i1 %cmp1.not.headerCopy.1, i1
  ... true, !dbg !51
  %67 = insertelement <vscale x 4 x i1> %66, i1 %cmp1.not.headerCopy.1.2, i1
  ... false, !dbg !51
  %68 = insertelement <vscale x 4 x i1> %67, i1 %cmp1.not.headerCopy.1.2.3, i1
  ... true, !dbg !51
  %69 = trunc i64 %12 to i32, !dbg !51
  %70 = call <vscale x 4 x i32> @llvm.aarch64.sve.index.nxv4i32(i32 %69, i32
  ... 1), !dbg !51
  br label %new.latch
```

```
new.latch:
  %72 = phi <vscale x 4 x i32> [ %70, %if.then ], [ %13, %linearized ]
  %73 = phi <vscale x 4 x i1> [ %68, %if.then ], [ %14, %linearized ]
  %74 = phi <vscale x 4 x i32> [ %59, %linearized ], [ %15, %if.then ]
  %75 = phi <vscale x 4 x i1> [ %57, %linearized ], [ %47, %if.then ]
  br label %for.body
```

```
linearized:
  %48 = getelementptr inbounds i32, ptr %a, i64 %12, !dbg !41
  %49 = load i32, ptr %48, align 4, !dbg !41, !tbaa !43
  %50 = getelementptr inbounds i32, ptr %b, i64 %12, !dbg !47
  %51 = load i32, ptr %50, align 4, !dbg !47, !tbaa !43
  %52 = mul nsw i32 %51, %49, !dbg !48
  %53 = getelementptr inbounds i32, ptr %c, i64 %12, !dbg !49
  store i32 %52, ptr %53, align 4, !dbg !50, !tbaa !43
  ... false
  %54 = insertelement <vscale x 4 x i1> undef, i1 %cmp1.not, i1 false
  %55 = insertelement <vscale x 4 x i1> %54, i1 %cmp1.not.headerCopy.1, i1 true
  ... true
  %56 = insertelement <vscale x 4 x i1> %55, i1 %cmp1.not.headerCopy.1.2, i1
  ... false
  %57 = insertelement <vscale x 4 x i1> %56, i1 %cmp1.not.headerCopy.1.2.3, i1
  ... true
  %58 = trunc i64 %12 to i32
  %59 = call <vscale x 4 x i32> @llvm.aarch64.sve.index.nxv4i32(i32 %58, i32 1)
  br label %new.latch
```